The Examiner objected to the disclosure stating that the computer programs in Appendices A and B are partially illegible. Accordingly, applicant will submit new copies of these programs upon allowance.

The Examiner rejected claim 4 under 35 USC Section 112, second paragraph. The Examiner indicated that in claim 4, line 12, "from" should be deleted, because "the status...has changed from between each said iterative evaluation" does not make sense. Accordingly, applicant has deleted the word "from" from this claim.

Finally, the Examiner indicated that claims 1-19 stand rejected and new claims 20-25 are rejected under 35 USC Section 102(b) as being clearly anticipated by Perkins et al.

Applicant claims a method for automatically evaluating a decisional rule containing a task and a condition which must be fulfilled before the task can be performed and for automatically performing the task when the condition is fulfilled. The method includes entering the decisional rule into computing means and compiling the decisional rule to parse the condition. The method also includes providing automatic and continuing iterative evaluations of whether the condition is fulfilled until the condition is fulfilled once. The method further includes automatically performing the task when the condition is fulfilled once and resuming further processing only after the condition is filled once.

See claim 1. This method is directed to a "when" type of decisional rule.

Claims 2-19 include a number of independent method claims similar to method claim 1 including the automatic and iterative evaluation, but which are directed to other types of decisional rules, such as, for example, "whenever" type decisional rules and a corresponding set of independent apparatus claims.

Claims 20-25 were added in applicant's preliminary amendment of May 24, 1995. Independent claim 20 is directed to a system including means for executing a decisional rule which performs a task upon the fulfillment of a condition. The system provides automatic and continuous iterative evaluations of whether a condition is fulfilled until the condition is fulfilled at least once and which resumes any further processing only after the condition is fulfilled once. The system is automatically programmed in order to achieve continuing recursive evaluation of the conditions contained within the decisional rule.

Independent claim 21 includes means for automatically iteratively evaluating whether the condition of the decisional rule is fulfilled, thereby eliminating the need for manually providing additional decisional rules which force sequential evaluation in programming loops. Independent claim 22 is a method claim which includes, *inter alia*, the step of automatically providing computer

instructions which accomplish automatic and continuous iterative evaluations of whether the condition is fulfilled. Independent claim 23 includes the step of automatically providing instructions which accomplish automatic and continuous iterative evaluations of whether the condition is fulfilled. Independent Claim 24 includes an executor means which includes means for providing automatic and continuing iterative evaluations of whether the condition is fulfilled.

Finally, independent method claim 25 is directed to a method for automatically processing a series of WHEN/THEN statements including control commands which automatically halt further processing until a condition associated with each control command is fulfilled. The method includes manually entering a sequence of WHEN/THEN statements including one or more control commands each containing a condition into processing means. The processing means automatically processes each statement sequentially in the sequence, automatically halts processing of the statement upon occurrence of one of the control commands, automatically and continuously provides iterative evaluation of whether the condition associated with the control command is fulfilled, and automatically resumes processing of the remainder of the statement in the sequence only upon the fulfillment of the condition associated with the control command.

involving many decisional rules. Astronauts on board the space shuttle may not have the skill and certainly do not have the time to write programs to accomplish tasks such as "when time = 15, initiate thrusters."

Perkins et al. relates to adding temporal reasoning to expert-system-building environments. Perkins et al. disclose a machine code program which evaluates the occurrence of a condition and performs a task if the condition occurs. However, the evaluation of the occurrence of the condition is repeated over and over again only because loops required to perform the recursive evaluations are part of the programming involved in setting up the decisional rule. That is, unlike applicant's claimed system and method, the user cannot simply input a condition and a task to be performed if the condition is met and have the system provide automatic and continuing iterative evaluations whether the condition is fulfilled until the condition is fulfilled. But rather, when entering the decisional rule, Perkins et al. require that a loop which provides for the recursive evaluation of the decisional rule be programmed into the decisional rule itself.

Perkins et al. describe a way of writing code. Applicant has invented a system which automatically writes code for the user. The user does not need to understand how to write code or loops. The system compiles the decisional rules, the system parses the condition, the system

Applicant's invention, as set forth in all of his claims, allow a user, such as an astronaut, who has little or no programming skill to enter a decisional rule containing, for example, only a condition and a task which is to be performed when the condition is met. Applicant's claimed system then automatically "constructs the loop" for recursively executing the decisional rule, e.g. "automatic and continuing iterative evaluations..." of claim 1. An object achieved by this invention is to provide a system and method which does not require manual programming of the system in order to achieve continuing recursive evaluation of conditions contained within a decisional rule.

As set forth in the Background of Invention of the subject application, decisional rules, such as IF-THEN statements are well known in computer programming. However, before the present invention, in order for these types of statements to be effective, computer programmers skilled in the art were required to program "loops" into their particular system so that recursive execution of the decisional rules could be carried out. Also as delineated in the Background of Invention, in many applications requiring the use of decisional rules, users may not be sufficiently skilled in computer science disciplines to program the system to meet their needs or, even if the users were sufficiently skilled such programming is time consuming, cumbersome and complex especially in systems

automatically provides continuing iterative evaluations of the rule, and the system resumes processing after the condition is fulfilled.

Perkins et al. describe a very different system, wherein the user writes code which is compiled, the user writes code which parses the condition, the user writes code which provides continuing iterative evaluations, the user writes code which performs a task when the condition is fulfilled, and the user writes code which resumes the processing after the condition is fulfilled. Thus, Perkins et al. do not disclose the applicant's claimed system which automatically evaluates a decisional rule.

Using the technique of Perkins, one cannot enter a command such as "when...then..." as a code into the computer. Instead, the user must write code which accomplishes the function of "when...then...". To do this, the user must understand computer programming and be able to write loops and nested loops if there are many conditions. The user must be able to keep track of nested loops, must exercise configuration control when changes are made, and must troubleshoot and debug the programs written, among other things.

Applicant's claimed invention does this for the user automatically. See the claims described above. For example, as set forth in independent claim 20, applicant's claimed system is automatically programmed in order to

achieve continuing recursive evaluation of the conditions contained within a decisional rule. Perkins et al. clearly teach a system which utilizes decisional rules; however, its rules are programmed into the system and when programmed include loops which enable the decisional rule to be recursively evaluated.

The Examiner states that the system described in the Perkins reference enables "a non-programmer to create an expert system by entering high level rules into the shell. The shell handles all other processing tasks. Perkins teaches the rule structure that is to be entered into the LES shell (see Figure 4 for the structure of the "when" portion of a rule, and figure 7 for an example of a complete WHEN-THEN rule)."

Although in Fig. 7 of Perkins et al., "WHEN:" and "THEN:" are used to illustrate the processing which occurs, Perkins et al. provide an English translation at the top of the figure which clearly shows that the process of this data-driven rule is actually an "IF-THEN" process. This IF-THEN process is not automatically recursively evaluated, but instead the programmer of this data-driven rule must provide the necessary loop construction in order to ensure that this loop is recursively evaluated.

The system of the Perkins et al. reference is designed to be used by "knowledge engineers" who are skilled in programming so that they may write decisional rules for

temporal-reasoning to be used in various applications. The "knowledge engineers" are required to construct the decisional rules needed for their particular applications including the instructions to recursively evaluate the decisional rules. The level of programming skill required of the users of the Perkins et al. system is much higher than that of applicant's system. Applicant's system enables persons unskilled in computer programming to enter relatively simplistic decisional rules and the system automatically recursively analyzes the rule. In contrast, the Perkins et al. system requires relatively skilled computer programming to construct decisional rules containing the structure to achieve recursive evaluation.

Each of Examiner's rejections has been addressed or traversed. Accordingly, it is respectfully submitted that the application is in condition for allowance. Early and favorable action is respectfully requested.

If for any reason this RESPONSE is found to be INCOMPLETE, or if at any time it appears that a TELEPHONE CONFERENCE with counsel would help advance prosecution, please telephone the undersigned or his associate, John W. Powell, collect in Waltham, Massachusetts, (617)890-5678.

Respectfully submitted,

Kirk Teska
Reg. No. 36,291